Listing of Claims

Claims 1-16 (canceled)

Claim 17 (currently amended) A method performed by a cryptographic device for generating an approximate authentication code, said method comprising the steps of:

a. receiving a message containing data and arranging the data into a table having $|A|$ columns and T^2 rows, where A and T are integers and T is selected to be an odd integer;

b. permuting at least some of the arranged data;

c. masking the permuted data;

d. copying the permuted and masked data into T S-arrays, each S-array having $|A|$ columns, and determining a majority bit value of each of the $|A|$ columns for each of the T S-arrays;

e. using the determined majority bits to create a T-array having $|A|$ columns and T rows and

f. determining the majority bit value of reach of the $|A|$ columns in the T array.

Claim 18 (original) The method of claim 17, further comprising the step of generating a pseudo-random bit string before the step of permuting.

Claim 19 (original) The method of claim 18, wherein the step of generating the pseudo-random bit string (PRBS) further comprises using a shared key and pseudo-random number generator to generate the PRBS.

Claim 20 (original) The method of claim 19, wherein the step of generating the PRBS further comprises using an initial value to generate the PRBS.

Claim 21 (canceled)

Claim 22 (original) The method of claim 17, further comprising the step of selecting a length of $|A|$.

Claim 23 (original) The method of claim 17, wherein the step of permuting comprises permuting the data by row.

Claim 24 (original) The method of claim 23, further comprising the step of for each permuted row, permuting data within each row.

Claim 25 (original) The method of claim 24, wherein the step of permuting data within each row further comprises the step of circularly shifting each permuted row a pseudo- random number of places.

Claim 26 (original) The method of claim, 17, wherein the step of permuting further comprises permuting the data by bit.

Claim 27 (original) The method of claim 17, wherein the step of permuting further comprises selecting an unpredictable permutation.

Claim 28 (original) The method of claim 27, wherein the step of selecting an unpredictable permutation further comprises using one of a block cipher and a conventional message authentication code.

Claim 29 (original) The method of claim 17, wherein the step of permuting comprises permuting all of the data in the message.

Claim 30 (original) The method of claim 17, wherein the step of permuting comprises permuting less than all of the data in the message.

Claim 31 (original) The method of claim 30, wherein the step of permuting further comprises using a pseudo-random function to select the data for permuting.

Claim 32 (original) The method of claim 30, wherein the step of permuting comprises permuting a random sample of data in the message.

Claim 33 (original) The method of claim 30, wherein the step of permuting further comprises permuting at least one of statistical data and averages of data in the message.

Claim 34 (original) The method of claim 17, wherein the step of masking further comprises the step of stream encrypting the permuted data.

Claim 35 (currently amended) The method of claim ~~17~~ 18, wherein the step of masking further comprises bitwise exclusive-ORing the permuted data and at least a portion of the pseudo-random bit ~~stream~~ string.

Claim 36 (original) The method of claim 17, wherein the step of masking comprises generating an unbiased, independent, identically distributed set of 1s and 0s.

Claim 37 (original) The method of claim 17 wherein the step of copying the permuted and masked data into S-arrays further comprises selecting each S-array to have T rows.

Claim 38 (currently amended) ~~The~~ A method ~~of claim 37 wherein,~~ performed by a cryptography device for generating an approximate message authentication code, said method comprising the steps of:

a. receiving a message containing data and arranging the data into a table having $|A|$ columns and T^2 rows where A and T are integers;

b. permuting at least some of the arranged data;

c. masking the permuted data;

d. copying the permuted and masked data into T S-arrays, each S-array having $|A|$ columns, and determining a majority bit value of each of the $|A|$ columns for each of the T S-arrays, the step of copying the permuted and masked data into S-arrays further ~~comprises~~ comprising selecting each S-array to have T rows and adding a row of pseudo-random bits to the S-array if T is an even number;

e. using the determined majority bits to create a T-array having $|A|$ columns and T rows; and

f. determining the majority bit value of each of the $|A|$ columns in the T array.

Claim 39 (original) The method of claim 17, wherein the step of copying the permuted and masked data into S-arrays further comprises not selecting each S-array to have the same number of rows.

Claim 40 (currently amended) A device for generating an approximate authentication code, comprising:

a. a pseudo- random string generator module configured to receive as input a secret key and to output a string of pseudo-random bits;

b. an arrangement module configured to receive a message containing data and arrange the data into a table having. $|A|$ columns and T^2 rows, where A and T are integers and T is an odd integer;

c. a permuting module responsive to the arranged data and at least a portion of the string of pseudo-random bits and configured to permute arranged data;

d. a masking module responsive to the permuting module and at last a portion of the string of pseudo-random bits and configured to mask the ~~randomized~~ permuted data; and

e. a majority module responsive to the masking module and configured to:

i. copy the masked data into T S-arrays, each array having $|A|$ columns and to determine the majority bit value of each of the $|A|$ columns of the S-arrays;

ii. use the determined majority bits to create a T array having $|A|$ columns and T rows; and

iii. determine the majority bit value of each of the $|A|$ columns in the T

array.

Claim 41 (original) The device of claim 40, wherein the pseudo-random bit string generator module further comprises a pseudo-random number generator.

Claim 42 (original) The device of claim 41, wherein the pseudo-random number generator is a cryptographically secure random number generator.

Claim 43 (currently amended) ~~The~~ A device ~~of claim 40,~~ for generating an approximate message authentication code, comprising

a. a pseudo-random bit string generator module configured to receive as input a secret key and to output a string of pseudo-random bits, the pseudo-random bit string generator module ~~is~~ being further configured to receive as input an initial value;

b. an arrangement module configured to receive a message containing data and arrange the data into a table having |A | columns and T^2 rows, where A and T are integers;

c. a permuting module responsive to the arranged data and at least a portion of the string of pseudo-random bits and configured to permute the arranged data;

d. a masking module responsive to the permuting module and at least a portion of the string of pseudo-random bits and configured to mask the permuted data; and

e. a majority module responsive to the masking module and configured to:

i. copy the masked data into T S-arrays, each array having and to determine the |A| columns, e majority bit value of each of the |A| columns for each of the DS-arrays;

ii. use the determined majority bits to create a T –array having | A |columns and T rows; and

iii. determine the majority bit value of the of the | A| columns in the T array; and

wherein the pseudo-random string generator module is further configured to receive as input an initial value.

Claim 44 (canceled)

Claim 45 (original) the device of claim 40, wherein |A | is selected to have a predetermined length, the selected length depending on a sensitivity to bit changes.

Claim 46 (original) the device of claim 40, wherein the permuting module is configured to permute the data by row.

Claim 47 (original) The device of claim 46, wherein the permuting module is further configured , for each permuted row, to permute the data within each row.

Claim 48 (original) The device of claim 47, wherein the permuting module is further configured to circularly shift each permuted row a pseudo-random number of places.

Claim 49 (original) The device of claim 40., wherein the permuting module is configured to permute the data by one bit, byte, and data word.

Claim 50 (original) The device of claim, 40, wherein the permuting module is configured to use a collision-free, unpredictable permutation to permute the data.

Claim 51 (original) The device of claim 40, wherein the permuting module is configured to permute all of the data in the message.

Claim 52 (original) The device of claim 40, wherein the permuting module is configured to permute less than all of the data in the message.

Claim 53 (previously amended) The device of claim 52, wherein the permuting module uses a pseudo-random function to select the data for permuting.

Claim 54 (original) The device of claim 52, wherein the permuting module is configured to permute a random sample of data in the message.

Claim 55 (original) The device of claim 52, wherein the permuting module is configured to permute at least one of statistical data and averages of data in the message.

Claim 56 (original) The device of claim 40, wherein the masking module further comprises an exclusive-OR circuit responsive to the permuting module and at least a portion of the string of pseudo-random bits.

Claim 57 (original) The device of claim 40, wherein the majority module is configured to copy the masked data into S-arrays each having the same number of rows.

Claim 58 (original) The device of claim 40, wherein the majority module is configured to copy the masked data into S-arrays not all having the same number of rows.

Claims 59-65 (canceled)

Amendments to the Drawings

The attached 10 sheets of formal drawings are being submitted, without any amendment thereto, to replace the informal drawings previously filed with this application.